AD-769 625

A NOTE ON COMPUTATIONAL STUDIES FOR SOLVING TRANSPORTATION PROBLEMS

Fred Glover, et al

Texas University

Prepared for:

Office of Naval Research

April 1973

Unclassified
Security Classification

AD 769625

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Center for Cybernetic Studies University of Texas at Austin	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

A Note on Computational Studies for Solving Transportation Problems

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

F. Glover, D. Karney, and Darwin Klingman

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
April 1973	~~18~~ 19	7

8a. CONTRACT OR GRANT NO. NR-047-021 b. PROJECT NO. N00014-67-A-0126-0008 c. d. N00014-67-A-0126-0009	9a. ORIGINATOR'S REPORT NUMBER(S) Center for Cybernetic Studies Research Report No. CS 134 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Office of Naval Research (Code 434) Department of The Navy Washington, D.C.

13. ABSTRACT

This note provides a mathematical explanation for the superiority of certain pivot criterion heuristics when using the Row-Column Sum Method to solve transportation problems. In addition, new pivot criteria are developed using this mathematical explanation which are shown to be computationally superior to the previously best pivot criteria.

ia

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Unclassified
Security Classification

A-31408

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Transportation Problems						
Networks						
Transhipment Problems						
Linear Programming						

ib

Research Report
CS 134

A NOTE ON COMPUTATIONAL STUDIES FOR SOLVING TRANSPORTATION PROBLEMS

by

Fred Glover*

David Karney**

Darwin Klingman***

April 1973

*Professor of Management Science, University of Colorado at Boulder
**Assistant Director of the Bureau of Business Research, University of Texas at Austin
***Associate Professor of Operations Research, Statistics, and Computer Sciences, University of Texas at Austin

This research was partly supported by a grant from the Farah Foundation and by ONR Contracts N00014-67-A-0126-0008 and N00014-67-A-0126-0009 with the Center for Cybernetic Studies, The University of Texas. Reproduction in whole or in part is permitted for any purpose of the United States Government.

CENTER FOR CYBERNETIC STUDIES
A. Charnes, Director
Business-Economics Building, 512
The University of Texas
Austin, Texas 78712

ABSTRACT

This note provides a mathematical explanation for the superiority of certain pivot criterion heuristics when using the Row-Column Sum Method to solve transportation problems. In addition, new pivot criteria are developed using this mathematical explanation which are shown to be computationally superior to the previously best pivot criteria.

ii

1. INTRODUCTION

Computational studies by Dennis [3], Srinivasan-Thompson [7], and Glover-Karney-Klingman-Napier [4] have tested different pivot criterion heuristics when using the Row-Column Sum Method [1] (often called the MODI method [2]) to solve both totally dense and nondense transportation problems. These studies have found two of these heuristic procedures to be uniformly best. One purpose of this note is to provide a mathematical explanation for this computationally derived result. Another purpose is to use this mathematical explanation to derive other pivot criteria which exploit all of the advantages of the two best pivot procedures in such a way that the search time to find the next pivot will be smaller. Computational comparisons are then provided in the last section. The results of this study show the superiority of one new criterion to the previously best pivot criterion.

The studies [3,4,7] tested different pivot criteria which scan the rows (origin nodes) of the transportation tableau one at a time until an improving cell is found. One of the pivot criteria tested (called the row first negative rule [4]) pivots the first encountered improving cell into the basis. Another criterion tested (called the modified row first negative rule [4]) scans the rows until it encounters the first row that contains an improving cell, and then selects the cell in this row which violates dual feasibility by the largest amount. Both of these pivot criteria resume scanning from the point at which they previously terminated. For instance, the row first negative rule begins searching at the cell following the "come-in cell" of the previous pivot; the modified first negative rule begins searching in the row following the row in which the last pivot occurred. (An improved place to resume the search is identified in Section 3.)

The principal theoretical result of this note is the following: the procedure of pivoting on improving cells associated with a given node until no such cells are left "normally" yields the same basis regardless of the order in which the pivots are made. This result is used to derive a pivot criterion that uses the "shortest route" (minimum number of pivots) to reach the indicated basis. This result further provides a mathematical explanation of the superiority of the modified first negative pivot criterion over the first negative rule.

2. NOTATION AND PROBLEM STATEMENT

We write the transportation problem in the form:

$$\text{Minimize} \quad x_{oo} = \sum_{\substack{i\varepsilon M \\ j\varepsilon N}} c_{ij}x_{ij} \tag{1}$$

$$\text{subject to:} \quad \sum_{j\varepsilon N} x_{ij} = a_i, \qquad i\varepsilon M = \{1,2,\ldots,m\} \tag{2}$$

$$\sum_{i\varepsilon M} x_{ij} = b_j, \qquad j\varepsilon N = \{1,2,\ldots,n\} \tag{3}$$

$$x_{ij} \geq 0, \qquad i\varepsilon M, \quad j\varepsilon N \tag{4}$$

where $\sum_{i\varepsilon M} a_i = \sum_{j\varepsilon N} b_j$.

Following standard terminology, the a_i parameters are called supplies and the b_j parameters are called demands. We associate these supplies and demands respectively with the rows and columns of an m x n transportation tableau whose cells contain the "cost coefficients" c_{ij}. In addition, the rows of the transportation tableau are referred to as origin nodes and the columns as destination nodes. The cell in row i and column j of the transportation tableau is referred to as cell (i,j) or arc (i,j). Lastly a set of m+n-1 cells is a <u>basis</u> if and only if it forms a spanning tree for the m+n nodes associated with the problem [1,2,6]. A cell (and its associated variable x_{ij}) is called <u>basic</u> if it is contained among those cells in the basis and is called nonbasic otherwise.

A basic solution is the unique assignment of the values to the x_{ij} variables satisfying equations (2) and (3) that result once each nonbasic x_{ij} has been set equal to zero. If such a solution satisfies (4) for all of the variables, then it is called primal feasible.

The dual problem to the transportation problem can be stated as:

$$\text{Maximize} \quad \sum_{i \in M} a_i R_i + \sum_{j \in N} b_j K_j \tag{5}$$

$$\text{subject to:} \quad R_i + K_j \leq c_{ij} \quad , \quad i \in M,\ j \in N \tag{6}$$

Corresponding to a particular basis of the primal problem is a set of "row multipliers" R_i and a set of "column multipliers" K_j (not unique) such that the "updated cost coefficient" π_{ij}, defined by $\pi_{ij} = c_{ij}-R_i-K_j$, is zero for all basic cells. A basic solution is dual feasible if in addition $\pi_{ij} \geq 0$ for all nonbasic variables x_{ij}. (The multipliers R_i and K_j on which the π_{ij} are based represent values assigned to the variables of the dual of the transportation problem.) Given a basic primal feasible solution then cell (i,j) is called an improving cell if $\pi_{ij} < 0$. By fundamental linear programming theory, a basic primal feasible solution is optimal if no improving cells exist.

The approach used to solve the transportation problem in the Row-Column Sum Method [1,2] is to start with a basic primal feasible solution and proceeds to pivot improving cells into the basis maintaining primal feasibility until no improving cells exist. An efficient way of storing and updating the basis and associated multiplier values is contained in [5]. The computational studies [3,4,7] present computational results using different ways of picking the improving cell (pivot criteria) to enter the basis. The purpose of this note is to explain the interrelationship of the pivot criteria tested in [3,4,7], to develop new pivot criteria, and present computational results on all of these pivot criteria.

3. MATHEMATICAL DEVELOPMENT

Given a basic primal feasible solution and associated multiplier values such that $c_{ij}-R_i-K_j=0$ for each basic cell (i,j), consider the problem of finding new (updated) multiplier values when cell (p,q) replaces cell (r,s) in the basis. Since any basis for a transportation problem is a spanning tree [1,2,6] deleting cell (r,s) from the current basis splits the basis graph into two disjoint trees T_r and T_s where T_r contains node r and T_s contains node s. Further the spanning tree property of a basis implies that cell (p,q) will reconnect these disjoint trees. However, the origin node p may be in either tree T_r or T_s. Thus we denote the tree containing node p by T_p and the tree containing node q by T_q, where one of the trees T_p and T_q is T_r and the other is T_s. These observations lead to the following Remark. (A similar result is given in [5].)

Remark 1

Updated (New) multiplier values R_i and K_j may be determined by setting

$K_j' = K_j + \pi_{pq}$ for all j in T_q

$R_i' = R_i - \pi_{pq}$ for all i in T_q

$R_i' = R_i$ for all i in T_p

$K_j' = K_j$ for all j in T_p

Proof:

The proof is based on the observation that it is possible to assign new values to multipliers R_i and K_j so that the multipliers associated with the nodes in T_p are unaltered. It follows that the updated costs associated with cells in T_p will likewise remain unaltered and consequently will retain the value zero. To offset this, the origin multiplier values associated with origin nodes in T_q must all be altered by $-\pi_{pq}$, whereupon the destination multiplier values in T_q must all be altered by π_{pq}. The validity of these changes is verified by noting that the updated cost associated with any basic cell (u,v) in T_q is $\pi'_{uv}=c_{uv}-R'_u-K'_v = c_{uv}-(R_u-\pi_{pq}) - (K_v+\pi_{pq}) = c_{uv}-R_u-K_v = 0$. In addition the updated cost associated with cell (p,q) is zero since $\pi'_{pq} = c_{pq}-R'_p-K'_q=c_{pq}-R_p-(K_q+\pi_{pq}) = c_{pq}-R_p-K_q-\pi_{pq}= 0$. Thus the new multiplier values yield updated cost coefficients which are equal to zero for all basic cells, as required.

This result implies that it is necessary to decrease the updated cost coefficients on a change of basis step only for those cells leading from origin nodes in T_q to destination nodes in T_p. Further, the amount of the decrease in each of these updated costs is precisely π_{pq}. It is very easy to be led by these facts to an erroneous conclusion. Specifically, it seems plausible to suppose that a good place to resume the search for an improving cell would be among the origin nodes in T_q. This is undeniably the case if the only improving cell associated with the current basis is cell (p,q). Then, any improving cells that exist after the change of basis must be associated with origin nodes in T_q. Logically, then the modified row first negative pivot criterion, which was found to be computationally best among the criteria tested [3,4,7], should be improved by changing its search criterion to begin with the origin nodes in T_q rather than with the node p+1. However, the computational results in Section 4 demonstrate that this is not the case.

Coupling Remark 1 with further observations. however, does lead to a rule which is superior to the best rule previously devised. By way of preliminary, note that Remark 1 also implies that the updated cost of a cell emanating from node p is unaltered if its destination node is in T_p. On the other hand, the updated cost is increased by $-\pi_{pq}$ if the destination node is in T_q. An immediate inference is that if the cells in a particular row of the transportation tableau are scanned sequentially and if the improving cells in this row are pivoted into the basis as they are encountered, this row will contain no improving cells once all the cells in this row have been examined. Thus, since the row first negative rule performs this type of scanning and pivoting procedure, once it has scanned a row, the row will contain no improving cells. An alternate pivot criterion, which would obtain the same result and which would embody the philosophical approach of the modified row first negative rule, is to scan a row of the transportation tableau, simultaneously creating a list of all the improving cells in this row and finding the most negative of these improving cells. The pivot criterion would then bring the most negative improving cell into the basis. It would then re-search the list, simultaneously culling out those cells whose updated costs are now non-negative and finding the most negative updated cost. Once the list has been exhausted, no improving cells exist in this row (due to the monotonic property of the updated costs), and the search for an improving cell should be resumed by searching the origins in the tree (T_q) associated with the destination node of the last arc (p,q) entering the basis. We shall call this pivot criterion the <u>revised row first negative rule.</u> If the search is resumed in the row following the pivot, we shall call this pivot criterion the <u>altered row first negative rule</u>. In Section 4 computational results are presented on these pivot criteria which demonstrate that the second of these is more efficient than any criteria previously tested.

We will now lay analytical groundwork to provide further explanation of the observed empirical results, and to pave the way for future analysis of other choice rules that may be devised. Our results also provide a mathematical explanation of the earlier findings [3,4,7] that the modified row first negative rule is superior to the row first negative rule. In particular, assume row p contains two improving cells (p,q) and (p,t). Consider the problem of deciding which cell to bring into the basis first if pivoting is to continue until neither of these cells are pivot eligible. Essentially this decision can be resolved by characterizing the "basis equivalent paths" associated with these nonbasic cells. (By the "basis equivalent path" of nonbasic cell (i,j), we mean the unique path of basic cells (arcs) connecting node i to node j.)

There are two possibilities for the basis equivalent paths associated with cells (p,q) and (p,t). Namely, the paths may be disjoint (i.e., the paths have no cells in common) or the paths may have some common cells. The following remarks identify the relevant conclusions for each case.

Remark 2

If the basis equivalent paths associated with (p,q) and (p,t) are disjoint, the order in which the cells are considered is unimportant - i.e., considering them in either order will result in the same amount of computation and ultimately yield the same basis.

Proof:

Observe (without loss of generality) that pivoting cell (p,q) into the basis will not alter the basis equivalent path associated with cell (p,t) since the cell leaving the basis will lie in the basis equivalent path associated with cell (p,q). Further, the only variables x_{ij} whose values are altered by this change of basis are those associated with the cells in the basis equivalent path of cell (p,q) and x_{pq}. Thus, the flow values x_{ij} associated with cells in the basis equivalent path of cell (p,t) are unaltered. Further the updated cost associated with cell (p,t) is unaltered since node t must lie in the same tree as node p (i.e., $t\epsilon T_p$) since a path exists from node t to node p when the cell leaving the basis during this change of basis operation is deleted. Therefore, cell (p,t) is still pivot eligible and its basis equivalent path is unaltered; consequently the same pivots will be performed regardless of the order in which cells (p,q) and (p,t) are brought into the basis and regardless of the order in which the pivots occur. Finally, the same basis will be attained after executing the two pivots (assuming nondegeneracy).

It is interesting to note that the foregoing remark characterizes an instance in which the pivots are not required to be performed sequentially, but can, in fact, be performed simultaneously or in parallel. This observation could be helpful when designing computer codes to exploit parallel processing computers. The following remark identifies the somewhat more complex relationships that hold when the basis equivalent paths are not disjoint.

Remark 3

Let cell (u,v) denote the cell leaving the basis if cell (p,q) is pivoted into the basis first and let cell (r,s) denote the cell leaving the basis if cell (p,t) is pivoted into the basis first. Let F_p denote the arcs simultaneously in the basis equivalent paths of cells (p,q) and (p,t), let F_t denote the arcs in the basis equivalent path of cell (p,t) that are not in F_p, and let F_q denote the arcs in the basis equivalent path of cell (p,q) that are not in F_p. Assume that $\pi_{pq} < \pi_{pt}$ and that the current solution is non-degenerate.

1) If the cells (u,v) and (r,s) are simultaneously in the basis equivalent paths of cells (p,q) and (p,t) (i.e., (u,v), (r,s)εF_p), then cells (u,v) and (r,s) are the same cell. Further, the most negative improving cell (p,q) should be pivoted into the basis first in order to minimize computational effort. If the most negative cell is pivoted first then only one pivot will result; pivoting in the other order will result in making two pivots. In either case, the final basis will be the same.

2) If (u,v)εF_p, (r,s)εF_t, then one pivot will result if cell (p,q) is pivoted into the basis first. If (p,t) is pivoted into the basis first, then two pivots will occur. Further, the bases will be different and the reduction in the objective function value will be largest if (p,q) is pivoted first.

3) If (u,v)εF_q, (r,s)εF_p, then two pivots will result regardless of the pivot order, and the same basis is obtained.

4) If (u,v)εF_q, (r,s)εF_t and there exists a cell (i,j)εF_p whose flow will be decreased during the pivot such that $x_{ij} < x_{ur} + x_{rs}$ then two pivots will result regardless of the pivot order. However, different bases will result and the reduction in the objective function value will be largest if cell (p,q) is pivoted first.

5) If (u,v)εF_q, (r,s)εF_t and for all cells (i,j)εF_p whose flow will be

decreased $x_{ij} \geq x_{ur}+x_{rs}$, then two pivots will be required regardless of the pivot order and the same basis will be the result.

Proof:

The proof of this remark is a straightforward application of the type of reasoning underlying the proofs of Remarks 1 and 2 but is quite lengthy and is therefore omitted.

Remark 3 indicates that, if the most negative improving cell is not pivoted into the basis first, then either extra computational effort may be required to obtain the same basis or a different basis having a lower objective function value may ultimately result. Further, in no case will pivoting on the less negative improving cell result in either a better objective function value or less computational work; thus, Remark 3 provides a partial explanation of the superiority of the modified row first negative rule. Also this remark reinforces the earlier arguments that the new pivot criteria are likely to be superior to the row first negative rule. In particular, one can verify using Remark 3 that successively pivoting on the most negative improving cells associated with a node will yield a basis which has no improving cells in this row in the fewest number of pivots; also this basis will have the most improved objective function value. Notice that this statement does not imply that the altered or revised row first negative rules are the best pivot criteria since it may not be optimal to eliminate all improving cells in a row before proceeding to another row. However, the computational results in the next section support the hypothesis that one of these pivot criterion heuristics is efficient.

4. COMPUTATIONAL RESULTS

In this section we present computational results on the modified row first negative rule (MRFN), the modified row first negative rule resuming the search in the subtree T_q (MRFN-T_q), the altered row first negative rule (ARFN), and the revised row first negative rule (RRFN). For each of these pivot criteria, total solution times in seconds and the number of pivots are given in Table 1 for 150 transportation problems varying in number of origins, destinations, and admissible cells. In addition for the MRFN-T_q criterion, statistics are given on the total number of rows considered while searching subtree T_q, (TNRC-T_q), the total number of pivots made while searching subtree T_q (TNPM-T_q) and total number of rows considered (TNRC) to find the optimal solution. Additional statistics given on the ARFN and RRFN rules are the total number of improving cells put in the list (TNIC) and the number of pivots made from the improving cells on the list (NPML).

The transportation problems used in the study were randomly generated using a uniform probability distribution. The total supply of each m x n transportation was set equal to 1000 m and the supply and demand amounts were picked using a uniform probability distribution. The cost coefficient of each admissible cell was between 1 and 100. For each problem size and number of arcs, five problems were generated and solved using the MRFN pivot rule. The problem with the median total solution time was then solved using the other pivot criteria. These statistics obtained from the median problems are reported in Table 1.

The CDC 6600 at The University of Texas at Austin Computation Center was used to solve the problems. Computer jobs were executed during periods when machine load was approximately the same. The transportation code used to solve the problems is the one reported in [4] and is 5-15 times faster than any widely available transportation code. The row minimum start rule [4] was used to find the starting solution for all pivot criteria. The code is written in FORTRAN and was executed on the CDC 6600 using the RUN compiler.

The results presented in Table 1 indicate that the ARFN pivot criteria is consistently better (with respect to total time) than the previously found best [3,4,7] pivot criterion, MRFN. This result is most interesting from a historical viewpoint since the study by Srinivasan-Thompson [7] tested a pivot criterion which resembles the ARFN rule except that it fails to exploit the monotonicity property of the improving costs within a row. In particular, the criterion of [7] introduced a variant of the modified row first negative rule whereby if the last pivot was associated with a cell in row p, the search for an improving cell was re-initiated at row p rather than starting at row p+1. Thus, this criterion successively scanned row p and pivoted into the basis the most negative improving cell. The monotonic property of the improving cells implies that this criterion ultimately performs the same pivots as the ARFN rule, but does so with a good deal of unnecessary effort. In particular, the criterion suffers two major drawbacks: (1) every cell in row p is searched at each iteration to find the most negative improving cell and (2) every cell in row p is searched once when row p contains no improving cells. As a result, this pivot rule was found to be inferior to the MRFN rule.

By contrast, the data in Table 1 indicate that the ARFN pivot criterion is the best among those tested with respect to total solution time. Interestingly, the MRFN rule remains the best with respect to total number of pivots, but the length of time required for its execution makes it a second runner to the ARFN rule. On the other hand, the subtree search procedure of the RRFN rule, which might seem a highly plausible candidate for reducing the total number of pivots and total solution time (as noted in the discussion following Remark 1), performed somewhat less impressively than the MRFN and the ARFN rules.

In fact, the subtree search substantially increased the number of pivots using both the MRFN-T_q and RRFN rules in many cases. Consequently the RRFN rule is the worst from both standpoints and the ARFN rule emerges as the new most efficient criterion for selecting pivot elements.

TABLE 1 - COMPUTATIONAL STATISTICS

ORIGINS	DESTINATIONS	Number of Arcs	MRFN Total Solution Time	MRFN Total Number of Pivots	MRFN-T_q Total Solution Time	MRFN-T_q Total Number of Pivots	MRFN-T_q TNRC-T_q	MRFN-T_q TNPT-T_q	MRFN-T_q TNRC	ARFN Total Solution Time	ARFN Total Number of Pivots	ARFN TNIC	ARFN NPML	RRFN Total Solution Time	RRFN Total Number of Pivots	RRFN TNIC	RRFN NPML
20	250	750	.671	112	.692	113	123	77	256	.677	139	485	63	.879	161	341	62
20	250	1000	.942	172	1.347	217	278	147	419	.944	203	728	89	1.318	230	467	72
20	250	1250	1.024	158	1.162	164	184	144	310	.965	168	697	66	1.313	207	743	67
20	250	1500	1.256	203	1.348	186	243	142	348	1.213	214	914	95	1.265	190	658	69
20	250	1750	1.432	191	1.469	179	218	131	334	1.348	233	1375	98	1.506	232	773	82
20	375	1125	1.246	166	1.501	180	215	99	352	1.270	185	637	67	1.486	217	501	73
20	375	1500	1.823	229	2.135	235	283	162	433	1.725	275	1255	131	2.064	305	855	109
20	375	1875	1.994	204	2.227	224	235	150	396	1.370	244	1747	106	1.996	417	705	80
20	375	2250	2.294	219	2.602	258	264	170	426	2.233	270	1601	107	2.689	3[illegible]3	1314	121
20	375	2625	2.546	226	3.206	300	317	190	522	2.267	265	2251	120	2.969	318	1618	118
10	500	750	1.337	82	1.478	82	129	69	248	1.280	97	335	33	.334	89	192	29
10	500	1000	1.763	116	1.982	122	106	51	228	1.635	118	915	44	1.660	120	648	41
10	500	1250	2.294	133	2.188	120	95	69	181	2.103	15[illegible]	1169	66	2.186	158	659	59
10	500	1500	2.318	114	2.844	161	132	94	247	2.122	123	1124	48	2.469	143	687	48
10	500	1750	2.900	162	2.254	98	86	57	143	2.566	192	1378	96	3.382	224	1348	73
25	500	1875	2.571	307	2.789	308	414	234	583	2.288	341	1559	149	2.701	348	980	125
25	500	2500	2.958	304	2.769	361	401	240	604	2.970	358	1821	168	3.311	370	1136	144
25	500	3125	3.634	318	3.569	273	323	199	480	2.943	318	2075	150	3.565	355	1249	144
25	500	3750	4.338	364	4.891	362	432	261	636	3.938	416	3604	184	4.667	487	1932	176
25	500	4375	4.501	328	5.335	346	395	235	596	4.169	392	3466	185	5.636	467	1798	168
25	600	2250	3.459	352	4.543	383	572	259	846	3.320	437	22[illegible]	214	4.503	500	1689	177
25	600	3000	4.008	358	5.210	385	514	287	768	3.945	[illegible]	3392	217	4.526	477	2299	175
25	600	3750	4.779	350	5.141	331	421	230	672	4.517	452	2884	204	5.508	425	2121	165
25	600	4500	5.117	331	5.661	359	431	255	635	4.368	363	2663	169	5.918	453	1919	157
25	600	5250	6.533	436	7.043	435	490	339	656	6.356	386	5683	308	7.396	569	3560	221
10	750	1125	2.605	124	2.802	132	117	72	240	2.401	131	672	58	2.997	171	599	40
10	750	1500	3.751	181	3.956	196	138	100	248	3.654	210	1290	[illegible]	3.949	220	915	50
10	750	1875	4.449	198	4.170	149	134	95	231	4.134	222	2397	99	4.532	229	1314	84
10	750	2250	5.411	222	6.542	281	215	174	421	5.003	268	3277	133	6.567	381	281	147
10	750	2625	6.083	217	6.544	212	151	115	285	5.474	237	3737	121	6.666	285	3411	103

References

1. Charnes, A. and W.W. Cooper, Management Models and Industrial Applications of Linear Programming Vols. I and II. New York: John Wiley and Sons, Inc., 1961.

2. Dantzig, G.B. Linear Programming and Extensions, Princeton, N.J.: Princeton University Press, 1963.

3. Dennis, J.B. "A High-Speed Computer Technique for the Transportation Problem," Journal of Association for Computing Machinery, Vol. 8, (1958), 132-153.

4. Glover, Fred, D. Karney, D. Klingman, and A. Napier, "A Computational Study on Start Procedures, Basis Change Criteria, and Solution Algorithms for Transportation Problems," To appear in Management Science.

5. Glover, Fred, D. Karney, and D. Klingman "The Augmented Predecessor Index Method For Locating Stepping-Stone Paths and Assigning Dual Prices in Distribution Problems," Transportation Science, 6,2, (1972), 171-179.

6. Simonnard, Michel, Linear Programming, Englewood, N.J.: Prentice-Hall, Inc., 1966.

7. Srinivasan, V. and G.L. Thompson, "Benefit-Cost Analysis of Coding Techniques for the Primal Transportation Algorithm," ACM, Vol. 20, (1973), 194-213.